Reasoning with Goals to Engineer Requirements

Colette Rolland

REASONING WITH GOALS TO ENGINEER REQUIREMENTS

Colette Rolland
Colette Rolland , Université de Paris 1, Panthéon Sorbonne 75013 Paris Cedex 13,
Email: rolland@univ-paris1.fr

Keywords: Systems goals, goal-driven requirements, goal-oriented requirements, requirements engineering, requirements elicitation Internet services, Dial-up networking

Abstract: The concept of a goal has been used in multiple domains such as management sciences and strategic planning, artificial intelligence and human computer interaction. Recently goal driven approaches have been developed and tried out to support requirements engineering activities such as requirements elicitation, specification, validation, modification, structuring and negotiation. The paper reviews various research efforts undertaken in this line of research. It uses L'Ecritoire, an approach which supports requirements elicitation, structuring and documenting as a basis to introduce issues in using goals to engineer requirements and to present the state-of-the art.

1. INTRODUCTION

Motivation for goal-driven requirements engineering (RE) : In (Lamsweerde, 2000), Axel van Lamsweerde defines RE (RE) as "concerned with the identification of goals to be achieved by the envisioned system, the operationalisation of such goals into services and constraints, and the assignment of responsibilities of resulting requirements to agents as humans, devices, and software". In this view, goals drive the RE process which focuses on goal centric activities such as goal elicitation, goal modelling, goal operationalisation and goal mapping onto software objects, events and operations.

Many authors will certainly agree to this position or to a similar one because goal driven approaches are seen today as a means to overcome the major drawback of traditional RE (RE) approaches that is, to lead to systems technically good but unable to respond to the needs of their users in an appropriate manner. Indeed, several field studies show that requirements misunderstanding is a major cause of system failure. For example, in the survey over 800 projects undertaken by 350 US companies which revealed that one third of the projects were never completed and one half succeeded only partially, poor requirements was identified as the major source of problems (Standish, 1995). Similarly, a recent survey over 3800 organisations in 17 European countries demonstrate that most of the perceived problems are related to requirements specification (>50%), and requirements management (50%) (ESI, 1996).

If we want better quality systems to be produced i.e. systems that meet the requirements of their users, RE needs to explore the objectives of different stakeholders and the activities carried out by them to meet these objectives in order to derive *purposeful system requirements*. Goal driven approaches aim at meeting this objective.

As shown in Figure 1, these approaches are motivated by establishing an *intentional relationship* between the *usage world* and the *system world* (Jarke and Pohl, 1993). The *usage world* describes the tasks, procedures, interactions etc. performed by agents and how systems are used to do work. It can be looked upon as containing the objectives that are to be met in the organisation and which are achieved by the activities carried out by agents. The *subject world*, contains knowledge of the real world domain about which the proposed system has to provide information. Requirements arise from both of these worlds. However, the subject world imposes domain- requirements which are facts of nature and reflect domain laws whereas the usage world generates user-defined requirements which arise from people in the organisation and reflect their goals, intentions and wishes. The *system world* is the world of system specifications in which the requirements arising from the other two worlds must be addressed.

These three worlds are interrelated as shown in Figure 1. User-defined requirements are captured by the *intentional relationship*. Domain-imposed requirements are captured by the *representation relationship*.

Understanding the *intentional relationship* is essential to comprehend the reason why a system should be constructed. The usage world provides the rationale for building a system. The purpose of developing a system is to be found outside the system itself, in the *enterprise,* or in other words, in the context in which the system will function. The relationship between the usage and system world addresses the issue of the system purpose and relates the system to the goals and objectives of the organisation. This relationship explains *why* the system is developed. Modelling this establishes the conceptual link between the envisaged system and its changing environment. *Goal-driven approaches* have been developed to address the semiotic, social link between the usage and the system world with the hope to construct systems that meet the needs of their organisation stakeholders.

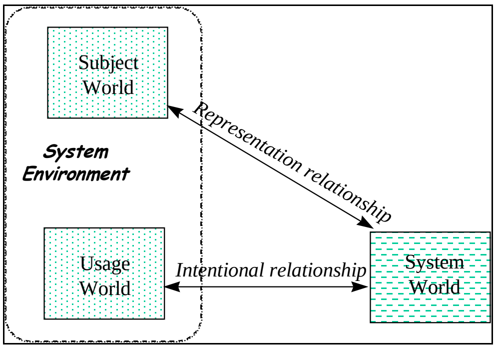

Figure 1: The relationships between the usage, subject and system worlds.

Roles of goal in RE: Goal modelling proved to be an effective way to *elicit requirements* (Potts, 1994; Rolland et al, 1998; Dardenne et al., 1993; Anton, 1994; Dubois et al., 1998; Kaindl, 2000; Lamsweerde, 2000). The argument of goal driven requirements elicitation being that the rationale for developing a system is to be found outside the system itself, in the enterprise (Loucopoulos, 1994) in which the system shall function.

RE assumes that the To-Be developed system might function and interact with its environment in many alternative ways. Alternative goal refinement proved helpful in the systematic *exploration of system choices* (Rolland et al, 1999; Lamsweerde, 2000; Yu, 1994).

Requirements completeness is a major RE issue. Yue (Yue, 1987) was probably the first to argue that goals provide a criterion for requirements completeness : the requirements specification is complete if the requirements are sufficient to achieve the goal they refine.

Goals provide a means to ensure *requirements pre-traceability* (Gotel et al., 1994; Pohl, 1996; Ramesh, 1995]. They establish a conceptual link between the system and its environment, thus facilitating the propagation of organisational changes into the system functionality. This link provides the rationale for requirements (Bubenko et al., 1994; Sommerville and Sawyer, 1997; Ross, 1977; Mostov, 1985; Yu, 1993) and facilitates the explanation and justification of requirements to the stakeholders.

Stakeholders provide useful and realistic viewpoints about the To-Be developed system but requirements engineers know that these viewpoints might be conflicting (Nuseibeh, 1994). Goals have been recognised to help in the *detection of conflicts* and their resolution (Lamsweerde, 2000; Robinson, 1989).

Difficulties with goal driven approaches : However, several authors (Lamsweerde et al., 1995; Anton, 1998; Rolland et al, 1998; Haumer et al, 1998) also acknowledge the fact that dealing with goal is not an easy task. We have applied the goal driven approach as embodied in the EKD method (Bubenko et al., 1994; Kardasis, 1998; Loucopoulos, 1997; Rolland et al., 1997b) to several domains, air traffic control, electricity supply, human resource management, tool set development. Our experience is that it is difficult for domain experts to deal with the fuzzy concept of a goal. Yet, domain experts need to discover the goals of real systems.

It is often assumed that systems are constructed with some goals in mind (Davis, 1993). However, practical experiences (Anton, 1996; ELEKTRA, 1997) show that goals are not given and therefore the question as to where they originate from (Anton, 1996) acquires importance. In addition, enterprise goals which initiate the goal discovery process do not reflect the actual situation but an idealised environmental one. Therefore, proceeding from this may lead to ineffective requirements (Potts, 1997). Thus, goal discovery is rarely an easy task.

Additionally, it has been shown (Anton, 1996) that the application of goal reduction methods (Dardenne et al., 1993) to discover the components goals of a goal, is not as straight-forward as literature suggests. Our own experience in the F3 (Bubenko et al., 1994) and ELEKTRA (Rolland et al., 1997a) projects is also similar. It is thus evident that help has to be provided so that goal modelling can be meaningfully performed.

Paper outline : The objective of this paper is (a) to highlight some of the issues of goal driven approaches in RE, (b) to provide an overview of the state-of-the art on these issues and (c) to illustrate how L'Ecritoire approach deals with them. In section 2 we briefly introduce L'Ecritoire, a goal driven approach developed in our group (Rolland et al, 1998; Tawbi, 2001; Ben Achour, 1999; Rolland et al, 1997b; Rolland et al, 1999) to support requirements elicitation, specification and documentation. The presentation of this approach in section 3 will be used as the means to raise issues in goal driven RE, and to provide a state-of-the art on these issues.

2. L'ECRITOIRE: AN OVERVIEW

L'Ecritoire is a tool for requirements elicitation, structuring, and documentation. Figure 2 shows that the approach underlying L'Ecritoire uses *goal-scenario coupling* to discover requirements from a computer-supported analysis of textual scenarios. L'Ecritoire produces a requirements document which relates system requirements (the functional & physical levels in Figure 2) to organisational goals (behavioural level in Figure 2).

Central to the approach is the notion of a *requirement chunk* (RC) which is a pair <goal, scenario>. A goal is 'something that some stakeholder hopes to achieve'(Plihon, 1998) whereas a scenario is a possible behaviour limited to a set of purposeful interactions taking place among agents'(CREWS, 1998). Since a goal is intentional and a scenario operational in nature, a RC is a possible way of achieving the goal.

L'Ecritoire aims at eliciting the collection of RCs through a *bi-directional coupling* of goals and scenarios allowing movement from goals to scenarios and vice-versa. As each goal is discovered, a scenario is authored for it. In this sense the goal-scenario coupling is exploited in the forward direction from goals to scenarios. Once a scenario has been authored, it is analysed to yield goals. This leads to goal discovery by moving along the goal-scenario relationship in the reverse direction. By exploiting the goal scenario relationship in the reverse direction, i.e. from scenario to goals, the approach proactively guides the requirements elicitation process.

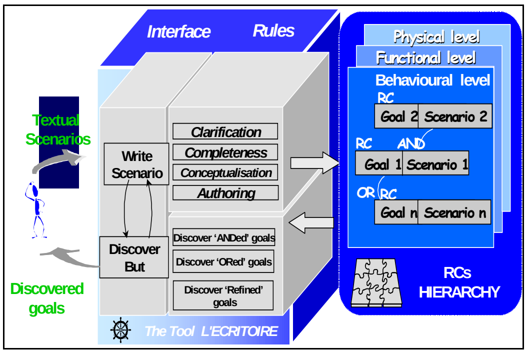

Figure 2: The L'Ecritoire architecture & functionality

The next section introduces the approach in more details with the aim to raise general issues reasoning with goals to engineer requirements and present the related state-of-the art. General issues are introduced with the ❖ symbol whereas the L'Ecritoire concepts are presented under the • symbol.

3. ISSUES IN GOAL REASONING

The notion of a goal is central to goal driven RE. In (Lamsweerde, 2001), a goal is an objective the system under consideration should achieve. Goals thus, refer to intended or *optative* (Jackson, 1995; Lamsweerde, 2001).

3.1 Goal formulation

• In L'Ecritoire, a goal is expressed as a clause with a main verb and several parameters, where each parameter plays a different role with respect to the verb. For example in the goal statement :

'Withdraw $_{verb}$ *(cash)* $_{target}$ *(from ATM)* $_{means}$*'*,

• *'Withdraw'* is the main verb, *'cash'* is the parameter target of the goal, and *'from ATM'* is a parameter describing the means by which the goal is achieved. We adopted the linguistic approach of Fillmore's Case grammar (Fillmore, 1968), and its extensions (Dik, 1989; Schank, 1973) to define goal parameters (Prat, 1997). Each type of parameter corresponds to a case and plays a different role with respect to the verb, e.g. target entities affected by the goal, means and manner to achieve the goal, beneficiary agent of the goal achievement, destination of a communication goal, source entities needed for goal achievement etc.

❖ Goal statements are often texts in natural language (Anton, 1996; Cockburn, 1995) and may be supplemented as suggested by (Zave, 1997) with an informal specification to make precise what the goal name designates.

The motivation for semi-formal or formal goal expressions is to be the support of some form of automatic analysis. We will see later in the paper how the L'Ecritoire goal template helps reasoning about goals. Typical semi-formal formulations use some goal taxonomy and associate the goal name to a predefined type (Anton, 1998; ELEKTRA, 1997; Dardenne et al., 1993).This helps clarifying the meaning of the goal. For instance, in (Mylopoulos, 1992) a non functional goal is specified by the specific sub-type it is instance of. Similarly, in Elektra (Elektra, 1997), goals for change are pre-fixed by one of the seven types of change: *Maintain, Cease, Improve, Add, Introduce, Extend, Adopt* and *replace*. Graphical notations (Chung et al., 2000; Mylopoulos, 1992; Lamsweerde, 2001) can be used in addition to a textual formulation.

Formal specifications of goals like in Kaos (Dardenne et al, 1993) require a higher effort but yield more powerful reasoning.

3.2 Coupling Goal and Scenario

• In L'Ecritoire, a *goal* is coupled with a *scenario*. In this direction, from goal to scenario, the relationship aims to concretise a goal through a scenario. Thus, the scenario represents a possible behaviour of the system to achieve the goal. In L'Ecritoire, a scenario is defined as composed of one or more *actions* which describe a unique path leading from an *initial* to a *final state* of agents. Below is an example of scenario associated to the goal '*Withdraw cash from the ATM*'.

> The user inserts a card in the ATM.
> The ATM checks the card validity.
> If the card is valid a prompt for code is given by the ATM to the user, the user inputs the code in the ATM.
> The ATM checks the code validity.
> If the code is valid, the ATM displays a prompt for amount to the user.
> The user enters an amount in the ATM.
> The ATM checks the amount validity.
> If the amount is valid, the ATM ejects the card to the user and then the ATM proposes a receipt to the user.
> The user enters the user's choice in the ATM.
> If a receipt was asked the receipt is printed by the ATM to the user but before the ATM delivers the cash to the user.

❖ Many authors suggest to combine goals and scenarios (Potts, 1995; Cockburn, 1995; Leite et al, 1997; Kaindl, 2000; Sutcliffe, 1998; Haumer et al., 1998; Anton, 1998; Lamsweerde et Willemet, 1998). (Potts, 1995) for example, says that it is « unwise to apply goal based requirements methods in isolation » and suggests to complement them with scenarios. This combination has been used mainly, to make goals concrete, i.e. to operationalise goals. This is because scenarios can be interpreted as containing information on how goals can be achieved. In (Dano et al., 1997; Jacobson, 1995; Leite, 1997; Pohl and Haumer, 1997), a goal is considered as a contextual property of a use case (Jacobson, 1995) i.e. a property that relates the scenario to its organisational context. Therefore, goals play a documenting role only. (Cockburn, 1995) goes beyond this view and suggests to use goals to structure use cases by connecting every action in a scenario to a goal assigned to an actor. In this sense a scenario is discovered each time a goal is. Clearly, all these views suggest a unidirectional relationship between goals and scenarios similarly to what we introduced in L'Ecritoire so far. We will see later on, how L'Ecritoire exploits the goal/scenario coupling in the reverse direction.

3.3 Relationships among Goals

• In L'Ecritoire, RCs can be assembled together through *composition, alternative and refinement* relationships. The first two lead to AND and OR structure of RCs whereas the last leads to the organisation of the collection of RCs as a hierarchy of chunks of different granularity.

AND relationships among RCs link complementary chunks in the sense that every one requires the others to define a completely functioning system. RCs linked through *OR relationships* represent alternative ways of fulfilling the same goal. RCs linked through a *refinement relationship* are at different levels of abstraction. The goal '*Fill in the ATM with cash'* is an example of *ANDed* goal *to 'Withdraw cash from the ATM'* whereas *'Withdraw cash from the ATM with two invalid code capture* 'is *ORed* to it. Finally *'Check the card validity'* is linked to the *goal 'Withdraw cash from the ATM'* by a *refinement* relationship.

❖ Many different types of *relationships* among goals have been introduced in the literature. They can be classified in two categories to relate goals: (1) to each other and (2) with other elements of requirements models. We consider them in turn.

AND/OR relationships (Bubenko et al, 1994; Dardenne et al, 1993; Rolland et al, 1998; Loucopoulos et al, 1997; Mylopoulos 1999) inspired from AND/OR graphs in Artificial Intelligence are used to capture goal decomposition into more operational goals and alternative goals, respectively. In the former, all the decomposed goals must be satisfied for the parent goal to be achieved whereas in the latter, if one of the alternative goals is achieved, then the parent goal is satisfied.

In (Mylopoulos, 1992; Chung et al., 2000), the inter-goal relationship is extended to support the capture of negative/positive influence between goals. A sub-goal is said to *contribute* partially to its parent goal. This leads to the notion of goal *satisfycing* instead of goal *satisfaction*. The 'motivates' and 'hinders' relationships among goals in (Bubenko et al, 1994) are similar in the sense that they capture positive/negative influence among goals.

Conflict relationships are introduced (Bubenko et al, 1994; Dardenne et al 1993; Nuseibeh, 1994; Easterbrook, 1994) to capture the fact that one goal might prevent the other to be satisfied.

In addition to inter-goal relationships, goals are also related to other elements of requirements models. As a logical termination of the AND/OR decomposition, goals link to operations which ensure them (Anton, 1994; Anton and Potts, 1998; Kaindl, 2000; Lamsweerde et Willemet, 1998). Relationships between goals and system objects have been studied in (Lee, 1997) and are inherently part of the KAOS model (Lamsweerde et al., 1991; Dardenne et al., 1993)).

Relationships with agents have been emphasized in (Yu 1993; Yu 1997) where a goal is the object of the dependency between two agents. Such type of link is introduced in other models as well (Dardenne et al, 1993; Lamweerde et al., 1991; Letier, 2001) to capture who is responsible of a goal. As discussed earlier, goals have been often coupled to scenarios (Potts, 1995; Cockburn, 1995; Leite, 1997; Kaindl, 2000; Sutcliffe, 1998; Haumer et al., 1998; Anton, 1998; . et al., 1998). In (Bubenko et al, 1994) goals are related to a number of concepts such as *problem*, *opportunity* and *thread* with the aim to understand better the context of a goal. Finally the interesting idea of *obstacle* introduced by (Potts, 1995) leads to obstructions and resolution relationships among goals and obstacles (Lamweerde, 2000a; Sutcliffe, 1998).

3.4 Levels of Abstraction

• The L'Ecritoire approach identifies three levels of requirements abstraction, namely the *behavioural*, *functional* and *physical* levels. The aim of the *behavioural level* is to couple the services that a system should provide so a business goal. At the *functional level* the focus is on the interactions between the system and its users to achieve the services assigned to the system at the behavioural level. The *physical level* focuses on what the system needs to perform the interactions selected at the system interaction level.

❖ As in L'Ecritoire goals many approaches suggest to formulate goals at different *levels of abstraction*. By essence goal centric approaches aim to help in the move from strategic concerns and high level goals to technical concerns and low abstraction level goals. Therefore, it is natural for approaches to identify different levels of goal abstraction where high level goals represent business objectives and are refined in system goals (Anton et al., 2001; Anton and Potts, 1998) or system constraints (Lamsweerde and Letier, 2000a). Inspired by cognitive engineering, some goal driven RE approaches deal with means-end hierarchy abstractions, where each hierarchical level represents a different model of the same system. The information at any level acts as a goal (the end) with respect to the model at the next lower level (the means) (Leveson 2000; Rasmussen, 1990; Vicente and Rasmussen, 1992).

3.5 Eliciting Goals

• The L'Ecritoire requirements elicitation process is organised around two main activities : *goal discovery* and *scenario authoring*.

In this process, *goal discovery* and *scenario authoring* are complementary activities, the former following the latter. As shown in Figure 3, these activities are repeated to incrementally populate the RCs hierarchy.

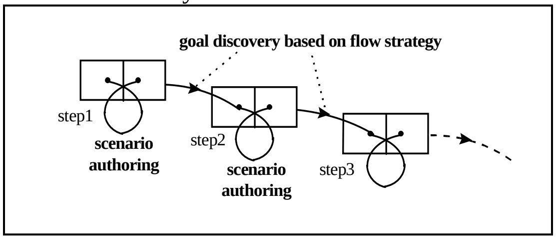

Figure 3: Goal reasoning in l'Ecritoire.

Each of the two main activities is supported by enactable rules, (1) *authoring rules* and (2) *discovery rules.* Authoring rules allow L'Ecritoire scenarios which are textual to be authored. Discovery rules are for discovering goals through the analysis of authored scenarios. We focus here on exemplifying the discovery rules. Details about the authoring rules and the linguistic approach underlying them can be found in (Rolland and Ben Achour, 1997; Ben Achour, 1999).

Discovery rules guide the L'Ecritoire user in discovering new goals and therefore, eliciting new RCs. The discovery is based on the analysis of scenarios through one of the three proposed discovery strategies, namely the *refinement, composition* and *alternative* strategies. These strategies correspond to the three types of relationships among RCs introduced above. Given a pair <G,Sc>:

- the *composition strategy* looks for goals Gi ANDed to G,
- the *alternative strategy* searches for goals Gj ORed to G,
- the *refinement strategy* aims at the discovery of goals Gk at a lower level of abstraction than G.

Once a complete scenario has been authored, any of these three strategies can be followed.

L'Ecritoire uses six discovery rules, two for each strategy. Rules can be applied at any of the three levels of abstraction, contextual, functional and physical. A detailed description of rules can be found in (Rolland et al., 1998; Tawbi, 2001, Rolland, 2002). As an example of a rule, we present the refinement rule R1 and exemplify it with the example of the ATM system engineering.

Refinement guiding rule (R1) :

Goal : Discover (from RC <G,Sc>)$_{So}$ (goals refined from G)$_{Res}$ (using every atomic action of Sc as a goal)$_{Man}$

Body :

1. *Associate a goal Gi to every atomic action Ai in Sc. Gi refines G*
2. *Complement Gi by the manner 'in a normal way'*
3. *User evaluates the proposed panel of goals Gi and selects the goals of interest*
4. *RCs corresponding to these selected goals are ANDed to one another*

The guiding rule R1 aims at refining a given RC *(from RC<G,Sc>)$_{So}$* by suggesting new goals at a lower level of abstraction than G *(goals refined from G)$_{Res}$*. The refinement mechanism underlying the rule looks to every interaction between two agents in the scenario Sc as a goal for the lower level of abstraction (step1). Let us take as an example the scenario SC associated to the goal *Improve services to our customers by providing cash from the ATM.*

Scenario SC :

1. *If the bank customer gets a card from the bank,*
2. *Then, the bank customer withdraws cash from the ATM*
3. *and the ATM reports cash transactions to the bank.*

This scenario includes three interactions namely 'Get card', 'Withdraw cash' and 'Report cash transactions' corresponding to the three services involving the ATM. These services are proposed as the three a finer grained goals :

- *'Get card from the bank in a normal way'*
- *'Withdraw cash from ATM in a normal way'*
- *'Report cash transactions to the bank in a normal way'*

Assuming that the user accepts the three suggested goals (step3), the corresponding RCs are ANDed to one another (step4).

❖ As illustrated above, L'Ecritoire develops a requirements/goal inductive elicitation technique based on the analysis of conceptualised scenarios. The conceptualisation of a scenario results of powerful analysis and transformation of textual scenarios using a linguistic approach based on a Case Grammar inspired by Fillmore's Case Theory (Fillmore, 1968) and its extensions (Dik, 1989; Schank, 1973). The pay-off of the scenario conceptualisation process is the ability to perform powerful induction on conceptualised scenarios. In (Lamweerde, 1998), a similar approach is developed that takes scenarios as examples and counter examples of the intended system behaviour and generates goals that cover positive scenarios and exclude the negative ones.

An obvious informal technique for finding goals is to systematically ask WHY and WHAT-IF questions (Potts et al, 1994), (Sutcliffe et al, 1998). In L'Ecritoire the refinement strategy helps discovering goals at a lower level of abstraction. This is a way to support goal decomposition. Another obvious technique to perform decomposition is to ask the HOW question (Lamsweerde et al., 1995). A heuristic based decomposition technique has been developed in (Loucopoulos et al., 1997) and (Letier, 2001).

An attempt to retrieved cases from a repository of process cases was developed in (Le, 1999). The software tool captures traces of RE processes using the NATURE contextual model (Nature, 1999) and develops a case based technique to retrieve process cases similar to the situation at hand.

4. CONCLUSION

Goal-driven RE was introduced mainly to provide the rationale of the To-Be system. Beyond this objective, we have seen that there are some other advantages :

- goals bridge the gap between organisational strategies and system requirements thus providing a conceptual link between the system and its organisational context;

- goal decomposition graphs provide the pre-traceability between high level strategic concerns and low level technical constraints; therefore facilitating the propagation of business changes onto system features;
- ORed goals introduce explicitly design choices that can be discussed, negotiated and decided upon;
- AND links among goals support the refinement of high level goals onto lower level goals till operationalisable goals are found and associated to system requirements;
- Powerful goal elicitation techniques facilitate the discovery of goal/requirements;
- Relationships between goals and concepts such as objects, events, operations etc. traditionally used in conceptual design facilitates the mapping of goal graphs onto design specification.

There are other advantages which flow from issues which were not verified with in the paper and that we sketch here :

- Goal-based negotiation is one of them (Boehm and In H, 1996).
- Conflict resolution is another one. (Nuseibeh, 1994) explains how conflicts arise from multiple view points and concerns and in (Lamsweerde et al., 1998a) various forms of conflict have been studied.
- Goal validation is a third one. (Sutcliffe et al, 1998) use a scenario generation technique to validate goal/requirement and in (Heymans and Dubois et al., 1998) the validation is based on scenario animation.
- Qualitative reasoning about goals is provided by the NFR framework (Mylopoulos, 1992; Chung et al, 2000).

5. REFERENCES

Anton, A. I., 1996,Goal based requirements analysis. Proceedings of the 2nd International Conference on Requirements Engineering ICRE'96, pp. 136-144.

Anton, A. I, and Potts C., 1998,The use of goals to surface requirements for evolving systems, International Conference on Software Engineering (ICSE `98) , Kyoto, Japan, pp. 157-166, 19-25 April 1998.

Anton, A. I., Earp J.B., Potts C., and Alspaugh T.A., 2001,The role of policy and stakeholder privacy values in requirements engineering, IEEE 5th International Symposium on Requirements Engineering (RE'01), Toronto, Canada, pp. 138-145, 27-31 August 2001.

Ben Achour, C., 1999,Requirements extraction from textual scenarios. PhD Thesis, University Paris6 Jussieu, January 1999.

Boehm, B., 1976,Software engineering. IEEE Transactions on Computers, 25(12): 1226-1241.

Boehm, B., 1996,Identify Quality-requirements conflicts, 1996, Proceedings ICRE, Second International Conference on Requirements Engineering, April 15-18, 1996, Colorado spring, Colorado, 218.

Bowen, T. P., Wigle, G. B., Tsai, J. T., 1985,Specification of software quality attributes. Report of Rome Air Development Center.

Bubenko, J., Rolland, C., Loucopoulos, P., de Antonellis V., 1994,Facilitating 'fuzzy to formal' requirements modelling. IEEE 1st Conference on Requirements Enginering, ICRE'94 pp. 154-158.

Cockburn, A., 1995,Structuring use cases with goals. Technical report. Human and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT.TR.95.1, http://members.aol.com/acocburn/papers/usecases.htm .CREWS Team, 1998,The crews glossary, CREWS report 98-1, http://SUNSITE.informatik.rwth-aachen.de/CREWS/reports.htm

Chung, K. L., Nixon B. A., and Yu E., Mylopoulos J., 2000,Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers.. 440 p.

Dano, B., Briand, H., and Barbier, F., 1997, A use case driven requirements engineering process. Third IEEE International Symposium On Requirements Engineering RE'97, Antapolis, Maryland, IEEE Computer Society Press.

Dardenne, A., Lamsweerde, A. v., and Fickas, S., 1993,Goal-directed Requirements Acquisition, Science of Computer Programming, 20, Elsevier, pp.3-50.

Davis, A. .M., 1993, Software requirements :objects, functions and states. Prentice Hall.

Dik, S. C., 1989, The theory of functional grammar, part i : the structure of the clause. Functional Grammar Series, Fories Publications.

Dubois, E., Yu, E., and Pettot, M., 1998, "From early to late formal requirements: a process-control case study". *Proc. IWSSD'98 – 9th International Workshop on software Specification and design*. Isobe.IEEE CS Press. April 1998, 34-42.

ELEKTRA consortium, 1997,Electrical enterprise knowledge for transforming applications. ELEKTRA Project Reports.

ESI96, European Software Institute, 1996,"European User survey analysis", Report USV_EUR 2.1, ESPITI Project, January 1996.

Fillmore, C., 1968,The case for case. In ''Universals in linguistic theory'', Holt, Rinehart and Winston (eds.),Bach & Harms Publishing Company, pp. 1-90.

Gote, O., and Finkelstein A., 1994,Modelling the contribution structure underlying requirements, in Proc. First Int. Workshop on Requirements Engineering : Foundation of Software Quality, Utrech, Netherlands.

Haumer, P., Pohl K., and Weidenhaupt K., 1998,Requirements elicitation and validation with real world scenes. IEEE Transactions on Software Engineering, Special Issue on Scenario Management, M. Jarke, R. Kurki-Suonio (eds.), Vol.24, N°12, pp.11036-1054.

Heymans, P., and Dubois, E.,.1998, Scenario-based techniques for supporting the elaboration and the validation of formal requirements. RE Journal, P. Loucopoulos, C. Potts (eds.), Springer, CREWS Deliverable N°98-30, http://SUNSITE.informatik.rwth-aachen.de/CREWS/

Jacobson, I., 1995,The Use case construct in object-oriented software engineering. In Scenario-Based__Design: Envisioning Work and Technology in System Development, J.M. Carroll (ed.), pp.309-336.

Jarke, M., and Pohl, K., 1993,Establishing visions in context: towards a model of requirements processes. Proc. 12th Intl. Conf. Information Systems, Orlando.

Kaindl, H., 2000, "A design process based on a model combining scenarios with goals and functions", IEEE Trans. on Systems, Man and Cybernetic, Vol. 30 No. 5, September 2000, 537-551.

Kardasis P., and Loucopoulos P., 1998,Aligning legacy information system to business processes. Submitted to CAiSE'98.

Lamsweerde, A. v., Dardenne, B., Delcourt, and F. Dubisy, 1991,"The KAOS project: knowledge acquisition in automated specification of software", Proc. AAAI Spring Symp. Series, Track: "Design of Composite Systems", Stanford University, March 1991, 59-62.

Lamsweerde, A. v., Dairmont, R., and Massonet, P., 1995,Goal directed elaboration of requirements for a meeting scheduler *: Problems and Lessons Learnt*, in Proc. Of RE'95 – 2nd Int. Symp. On Requirements Engineering, York, pp 194 –204.

Lamsweerde A. v., and Willemet, L., 1998, "Inferring declarative requirements specifications from operational scenarios". In: IEEE Transactions on Software Engineering, Special Issue on Scenario Management. Vol. 24, No. 12, Dec. 1998, 1089-1114.

Lamsweerde, A. v., Darimont, R., and Letier, E., 1998a, "Managing conflicts in goal-driven requirements engineering", IEEE Trans. on Software. Engineering, Special Issue on Inconsistency Management in Software Development, Vol. 24 No. 11, November 1998, 908-926.

Lamsweerde, A. v., 2000,Requirements engineering in the year 00*: A research perspective*. In Proceedings 22nd International Conference on Software Engineering, Invited Paper, ACM Press, June 2000.
Lamsweerde, A. v., and Letier, E., 2000a,"Handling obstacles in goal-oriented requirements engineering", IEEE Transactions on Software Engineering, Special Issue on Exception Handling, Vol. 26 No. 10, October 2000, pp. 978-1005.
Lamsweerde, A.v., 2001, "Goal-oriented requirements engineering: a guided tour". Invited minitutorial, Proc. RE'01 International Joint Conference on Requirements Engineering, Toronto, IEEE, August 2001, pp.249-263.
Le, T. .L., 1999,Guidage des processus d'ingénierie des besoins par un approche de réutilisation de cas, Master Thesis, CRI, Université Paris-1, Panthéon Sorbonne.
Lee, S. P., 1997, Issues in requirements engineering of object-oriented information system: a review, Malaysian Journal of computer Science, vol. 10, N° 2, December 1997.
Leite, J. C. .S., Rossi, G., Balaguer, F., Maiorana, A., Kaplan, G., Hadad, G., and Oliveros, A., 1997, Enhancing a requirements baseline with scenarios. In Third IEEE International Symposium On Requirements Engineering RE'97, Antapolis, Maryland, IEEE Computer Society Press, pp. 44-53.
Letier, E., 2001,Reasoning about agents in goal-oriented requirements engineering. Ph. D. Thesis, University of Louvain, May 2001; http://www.info.ucl.ac.be/people/eletier/thesis.html
Leveson, N. G., 2000,"Intent specifications: an approach to building human-centred specifications", IEEE Trans. Soft. Eng., vol. 26, pp. 15-35.
Loucopoulos, P, 1994, The f^3 (from fuzzy to formal) view on requirements engineering. Ingénierie des systèmes d'information, Vol. 2 N° 6, pp. 639-655.
Loucopoulos, P., Kavakli, V., and Prakas, N., 1997,Using the EKD approach, the modelling component. ELEKTRA project internal report.
Mylopoulos, J.,. Chung K.L., and Nixon, B.A., 1992,Representing and using non- functional requirements: a process-oriented approach . IEEE Transactions on Software Engineering, Special Issue on Knowledge Representation and Reasoning in Software Development, Vol. 18, N° 6, June 1992, pp. 483-497.
Mylopoulos, J.,. Chung, K..L., and Yu, E., 1999,"From object-oriented to goal6oriented requirements analysis". Communications of the ACM. Vol 42 N° 1, January 1999, 31-37.
Mostow, J., 1985, "Towards better models of the design process". AI Magazine, Vol. 6, pp. 44-57.
Nature, 1999, The nature of requirements engineering . Shaker Verlag Gmbh. (Eds.) Jarke M., Rolland C., Sutcliffe A. and Dömges R., .Jily 1999.
Nuseibeh, B., Kramer, J., and Finkelstein, A., 1994,A framework for expressing the relationships between multiple views in requirements specification. In IEEE Transactions on Software Engineering, volume 20, pages 760-- 773. IEEE CS Press, October 1994.
Plihon, V., Ralyté, J., Benjamen, A., Maiden, N. A. M., Sutcliffe, A., Dubois, E., and Heymans, P., 1998, A reuse-oriented approach for the construction of scenario based methods. Proceedings of the International Software Process Associations 5th International Conference on Software Process (ICSP'98), Chicago.
Pohl K., 1996, Process centred requirements engineering, J. Wiley and Sons Ltd.
Pohl K., and Haumer, P., 1997, Modelling contextual information about scenarios. Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'97, Barcelona, pp.187-204, June 1997.
Potts, C., Takahashi, K., and Anton, A. I., 1994,Inquiry-based requirements analysis. In IEEE Software 11(2), pp. 21-32.
Potts, C., 1995, "Using schematic scenarios to understand user needs", Proc. DIS'95 - ACM Symposium on Designing interactive Systems: Processes, Practices and Techniques, University of Michigan, August 1995.
Potts, C., 1997,Fitness for use : the system quality that matters most. Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'97 , Barcelona, pp. 15-28, June 1997.
Prat, N., 1997,Goal formalisation and classification for requirements engineering. Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'97, Barcelona, pp. 145-156, June 1997.
Ramesh, B., Powers, T., Stubbs, C., and Edwards, M., 1995, Implementing requirements traceability : a case study, in Proceedings of the 2nd Symposium on Requirements Engineering (RE'95), pp89-95, UK.
Rasmussen, J., 1990, Mental models and the control of action in complex environments. Mental Models and Human--Computer Interaction, D. Ackermann and M.J. Tauber , eds., North-Holland : Elsevier, pp. 41-69.
Rolland, C, and Ben Achour, C., 1997,Guiding the construction of textual use case specifications. Data & Knowledge Engineering Journal Vol. 25 N° 1, pp. 125-160, (ed. P. Chen, R.P. van de Riet) North Holland, Elsevier Science Publishers. March 1997..
Rolland, C., Grosz, G., and Nurcan, S., 1997a,Guiding the EKD process. ELEKTRA project report,
Rolland, C., Nurcan, S., and Grosz, G., 1997b,Guiding the participative design process. Association for Information Systems Americas Conference, Indianapolis, Indiana, pp. 922-924, August, 1997
Rolland, C., Souveyet, C., and Ben Achour, C., 1998,Guiding goal modelling using scenarios. IEEE Transactions on Software Engineering, Special Issue on Scenario Management, Vol. 24, No. 12, Dec. 1998**.**
Rolland, C., 2002,L'e-lyee: l'ecritoire and lyeeall, Information and software Technology 44 (2002) 185-194
Rolland, C., Grosz, G., and Kla, R., 1999,Experience with goal-scenario coupling. in requirements engineering, Proceedings of the Fourth IEEE International Symposium on Requirements Engineering, Limerik, Ireland,
Ross, D. T., and Schoman, K. .E., 1977,Structured analysis for requirements definition. IEEE Transactions on Software Engineering , vol. 3, N° 1, , 6-15.
Robinson, W. N., 1989, "integrating multiple specifications using domain goals", Proc. IWSSD-5 - 5th Intl. Workshop on Software Specification and Design, IEEE, 1989, 219-225.
Schank, R. C., 1973,Identification of conceptualisations underlying natural language. In ''Computer models of thought and language'', R.C. Shank, K.M. Colby (Eds.), Freeman, San Francisco, pp. 187-247.
Sommerville I.,1996, Software Engineering. Addison Wesley.
Sommerville, I., and Sawyer, P., 1997, Requirements engineering. Worldwide Series in Computer Science, Wiley.
Standish Group, 1995,Chaos. Standish Group Internal Report, http://www.standishgroup.com/chaos.html
Sutcliffe, A.G., Maiden, N. .A., Minocha, S., and Manuel D.., 1998,"Supporting scenario-based requirements engineering", IEEE Trans. Software Eng. vol. 24, no. 12, Dec.1998, 1072-1088.
Tawbi, M., 2001,Crews-L'Ecritoire : un guidage outillé du processus d'Ingénierie des Besoins. Ph.D. Thesis University of Paris 1, October 2001.
Thayer, R., Dorfman, M. (eds.), System and software requirements. IEEE Computer Society Press.1990.
Vicente, K. J., and Rasmussen, J., 1992, Ecological interface design: Theoretical foundations. IEEE Trans. on Systems, Man, and Cybernetics, vol. 22, No. 4, July/August 1992.
Yu, E., 1994,Modelling strategic relationships for process reengineering. Ph.D. Thesis, Dept. Computer Science, University of Toronto, Dec. 1994.
Yue, K., 1987,What does it mean to say that a specification is complete?, Proc. IWSSD-4. Four International Workshop on Software Specification and Design, Monterrey, 1987.
Zave P., and Jackson M., 1997, "Four dark corners of requirements engineering", ACM Transactions on Software Engineering and Methodology, 1-30. 1997.